

Python: module vcs.isoline

vcs.isoline

[index](#)

```
# Isoline (Gi) module
#
#####
#
# Module:      isoline (Gi) module
#
# Copyright:   2000, Regents of the University of California
#              This software may not be distributed to others without
#              permission of the author.
#
# Author:      PCMDI Software Team
#              Lawrence Livermore NationalLaboratory:
#              support@pcmdi.llnl.gov
#
# Description: Python command wrapper for VCS's isoline graphics method.
#
# Version:     4.0
#
#####
#
#
#####
#
# Import: VCS C extension module.
#
#####
```

Modules

vcs.Canvas	vcs. vcs	vcs.queries
vcs.VCS_validation_functions	cdtime	vcs

Classes

[__builtin__.object](#)
[Gi](#)

class [Gi](#)([__builtin__.object](#))

Class: Gi # Isoline

Description of Gi Class:

The Isoline graphics method draws lines of constant value at specified levels in order to graphically represent a two-dimensional array. It also labels the values of these isolines on the VCS Canvas. The example below shows how to plot isolines of different types at specified values and how to create isoline labels having user-specified text and font size and color.

This class is used to define an isoline table entry used in VCS, and can be used to change some or all of the isoline attributes in an existing table entry.

Other Useful Functions:

```
a=vcs.init() # Constructor
a.show('isoline') # Show predefined isoline graphics
a.show('line') # Show predefined VCS line of type 'line'
a.setcolormap("AMIP") # Change the VCS color map
a.isoline(s,a,'default') # Plot data 's' with isoline using
                        # 'default' template
a.update() # Updates the VCS Canvas at the current time
a.mode=1, or 0 # If 1, then automatic updates; if
                # 0, then use update function to
                # update the VCS Canvas.
```

Example of Use:

```
a=vcs.init()
To Create a new instance of isoline use:
iso=a.createisoline('new','quick') # Copies content of 'quick'
iso=a.createisoline('new') # Copies content of 'default'
```

To Modify an existing isoline use:

```
iso=a.getisoline('AMIP_psl')
```

```
iso.list() # Will list all the isoline entries
iso.projection='linear'
lon30={-180:'180W',-150:'150W',0:'Eq'}
iso.xticlabels1=lon30
iso.xticlabels2=lon30
iso.xticlabels(lon30, lon30) # Will set them both
iso.xmtics1=''
iso.xmtics2=''
iso.xmtics(lon30, lon30) # Will set them both
iso.yticlabels1=lat10
iso.yticlabels2=lat10
iso.yticlabels(lat10, lat10) # Will set them both
iso.ymtics1=''
iso.ymtics2=''
iso.ymtics(lat10, lat10) # Will set them both
iso.dataawc_y1=-90.0
iso.dataawc_y2=90.0
```

```

iso.datawc_x1=-180.0
iso.datawc_x2=180.0
iso.datawc(-90, 90, -180, 180)      # Will set them all
axisconvert='linear'
yaxisconvert='linear'
iso.xyscale('linear', 'area_wt')    # Will set them both

```

There are many possibilities ways to set the isoline values:

A) As a list of tuples (Examples):

```

iso.level=[(23,32,45,50,76),]
iso.level=[(22,33,44,55,66)]
iso.level=[(20,0.0),(30,0),(50,0)]
iso.level=[(23,32,45,50,76),(35,45,55)]

```

B) As a tuple of lists (Examples):

```

iso.level=( [23,32,45,50,76], )
iso.level=( [22,33,44,55,66] )
iso.level=( [23,32,45,50,76], )
iso.level=( [0,20,25,30,35,40], [30,40], [50,60] )

```

C) As a list of lists (Examples):

```

iso.level=[[20,0.0],[30,0],[50,0]]

```

D) As a tuple of tuples (Examples):

```

iso.level=((20,0.0),(30,0),(50,0),(60,0),(70,0))

```

Note: a combination of a pair (i.e., (30,0) or [30,0]) represents the isoline value plus its increment value. Thus, to generate "default" isolines enter the following:

```

iso.level=[[0,1e20]]      # Same as iso.level=((0,1e20))

```

Displaying isoline labels:

```

iso.label='y'              # Same as iso.label=1, will show labels
iso.label='n'              # Same as iso.label=0, will not show labels

```

Specify the isoline line style (or type):

```

iso.line=( [0,1,2,3,4] )   # Same as iso.line='solid'
iso.line=( ['solid', 'dash', 'dot', 'dash-dot', 'long-dash'] ),
                    specify the isoline style

```

There are three possibilities for setting the line color indices:

```

iso.linecolors=(22,33,44,55,66,77)      # Same as iso.linecolors=[22,33,44,55,66,77]
iso.linecolors=( [22,33,44,55,66,77] )  # Will set the isoline color index
iso.linecolors=None                       # Turns off the line color index

```

There are three possibilities for setting the line widths (Ex):

```

iso.linewidths=(1,10,3,4,5,6,7,8)      # Same as iso.linewidths=[1,10,3,4,5,6,7,8]
iso.linewidths=( [1,2,3,4,5,6,7,8] )   # Will set the isoline width size
iso.linewidths=None                     # Turns off the line width size

```

There are three ways to specify the text or font number:

```

iso.text=(1,2,3,4,5,6,7,8,9)          # Font numbers are between 1 and 9
iso.text=[9,8,7,6,5,4,3,2,1]

```

```
iso.text=( [1, 3, 5, 6, 9, 2])
iso.text=None # Removes the text settings
```

There are three possibilities for setting the text color indices

```
iso.textcolors=( [22, 33, 44, 55, 66, 77])
iso.textcolors=(16, 19, 33, 44)
iso.textcolors=None # Turns off the text color i
```

Methods defined here:

```
__init__(self, parent, Gi_name=None, Gi_name_src='default', createGi=0)
```

```
datawc(self, dsp1=1e+20, dsp2=1e+20, dsp3=1e+20, dsp4=1e+20)
```

```
list(self)
```

```
rename = renameGi(self, old_name, new_name)
```

```
#####
#
# Function:      renameGi
#
# Description of Function:
#     Private function that renames the name of an existing
#     graphics method.
#
#
# Example of Use:
#     renameGi(old_name, new_name)
#         where: old_name is the current name of isoline
#         new_name is the new name for the isolin
#
#####
```

```
script(self, script_filename=None, mode=None)
```

```
Function:      script # Calls _vcs.s
```

```
Description of Function:
    Saves out a isoline graphics method in Python and VCS s
    designated file.
```

Example of Use:

```
script(scriptfile_name, mode)
    where: scriptfile_name is the output name of the
           mode is either "w" for replace or "a" for
```

Note: If the the filename has a ".py" at the end
 Python script. If the filename has a ".scr
 produce a VCS script. If neither extension
 default a Python script will be produced.

```
a=vcs.init()
```

```
iso=a.createisoline('temp')
iso.script('filename.py')
iso.script('filename.scr')
iso.script('filename','w')
```

```
# Append to a Python file
# Append to a VCS file "
```

xmtics(self, xmt1=", xmt2=")

xticlabels(self, xtl1=", xtl2=")

xyscale(self, xat=", yat=")

ymtics(self, ymt1=", ymt2=")

yticlabels(self, ytl1=", ytl2=")

Properties defined here:

datawc_calendar

```
get">get = _getcalendar(self)
set">set = _setcalendar(self, value)
```

datawc_timeunits

```
get">get = _gettimeunits(self)
set">set = _settimeunits(self, value)
```

datawc_x1

```
get">get = _getdatawc_x1(self)
set">set = _setdatawc_x1(self, value)
```

datawc_x2

```
get">get = _getdatawc_x2(self)
set">set = _setdatawc_x2(self, value)
```

datawc_y1

```
get">get = _getdatawc_y1(self)
set">set = _setdatawc_y1(self, value)
```

datawc_y2

```
get">get = _getdatawc_y2(self)
set">set = _setdatawc_y2(self, value)
```

label

```
get">get = _getlabel(self)
set">set = _setlabel(self, value)
```

level

```
get">get = _getlevels(self)
set">set = _setlevels(self, value)
```

levels

```
get">get = _getlevels(self)
set">set = _setlevels(self, value)
```

line

```
get">get = _getline(self)  
set">set = _setline(self, value)
```

linecolors

```
get">get = _getlinecolors(self)  
set">set = _setlinecolors(self, value)
```

linewidths

```
get">get = _getlinewidths(self)  
set">set = _setlinewidths(self, value)
```

name

```
get">get = _getname(self)  
set">set = _setname(self, value)
```

projection

```
get">get = _getprojection(self)  
set">set = _setprojection(self, value)
```

text

```
get">get = _gettext(self)  
set">set = _settext(self, value)
```

textcolors

```
get">get = _gettextcolors(self)  
set">set = _settextcolors(self, value)
```

xaxisconvert

```
get">get = _getxaxisconvert(self)  
set">set = _setxaxisconvert(self, value)
```

xmtics1

```
get">get = _getxmtics1(self)  
set">set = _setxmtics1(self, value)
```

xmtics2

```
get">get = _getxmtics2(self)  
set">set = _setxmtics2(self, value)
```

xticlabels1

```
get">get = _getxticlabels1(self)  
set">set = _setxticlabels1(self, value)
```

xticlabels2

```
get">get = _getxticlabels2(self)  
set">set = _setxticlabels2(self, value)
```

yaxisconvert

```
get">get = _getyaxisconvert(self)  
set">set = _setyaxisconvert(self, value)
```

ymtics1

```
get">get = _getymtics1(self)
set">set = _setymtics1(self, value)
```

ymtics2

```
get">get = _getymtics2(self)
set">set = _setymtics2(self, value)
```

yticlabels1

```
get">get = _getyticlabels1(self)
set">set = _setyticlabels1(self, value)
```

yticlabels2

```
get">get = _getyticlabels2(self)
set">set = _setyticlabels2(self, value)
```

Data and other attributes defined here:

```
__slots__ = ['setmember', 'parent', 'name', 'g_name', 'xaxisconvert', 'yaxisconvert', 'levels', 'level', 'la  
'text', 'textcolors', 'projection', 'xticlabels1', 'xticlabels2', 'yticlabels1', 'yticlabels2', 'xmtics1', ...]
```

g_name = <member 'g_name' of 'Gi' objects>

parent = <member 'parent' of 'Gi' objects>

setmember = <member 'setmember' of 'Gi' objects>

Functions

```
getGimember(self, member)
```

```
#####  
#  
# Function:      getGimember  
#  
# Description of Function:  
#     Private function that retrieves the isoline members from t  
#     structure and passes it back to Python.  
#  
#  
# Example of Use:  
#     return_value =  
#     getGimember(self, name)  
#         where: self is the class (e.g., Gi)  
#         name is the name of the member that is being  
#  
#####
```

```
getmember = getGimember(self, member)
```

```
#####
```

```

#
# Function:      getGimember
#
# Description of Function:
#     Private function that retrieves the isoline members from t
#     structure and passes it back to Python.
#
#
# Example of Use:
#     return_value =
#     getGimember(self, name)
#         where: self is the class (e.g., Gi)
#                name is the name of the member that is being
#
#####

```

renameGi(self, old_name, new_name)

```

#####
#
# Function:      renameGi
#
# Description of Function:
#     Private function that renames the name of an existing isol
#     graphics method.
#
#
# Example of Use:
#     renameGi(old_name, new_name)
#         where: old_name is the current name of isoline grap
#                new_name is the new name for the isoline gra
#
#####

```

setGimember(self, member, value)

```

#####
#
# Function:      setGimember
#
# Description of Function:
#     Private function to update the VCS canvas plot. If the can
#     set to 0, then this function does nothing.
#
#
# Example of Use:
#     setGimember(self, name, value)
#         where: self is the class (e.g., Gi)
#                name is the name of the member that is being
#                value is the new value of the member (or att
#
#####

```

setmember = setGimember(self, member, value)



```
#####  
#  
# Function:      setGimember  
#  
# Description of Function:  
#     Private function to update the VCS canvas plot. If the can  
#     set to 0, then this function does nothing.  
#  
#  
# Example of Use:  
#     setGimember(self,name,value)  
#           where: self is the class (e.g., Gi)  
#                   name is the name of the member that is being  
#                   value is the new value of the member (or att  
#  
#####
```